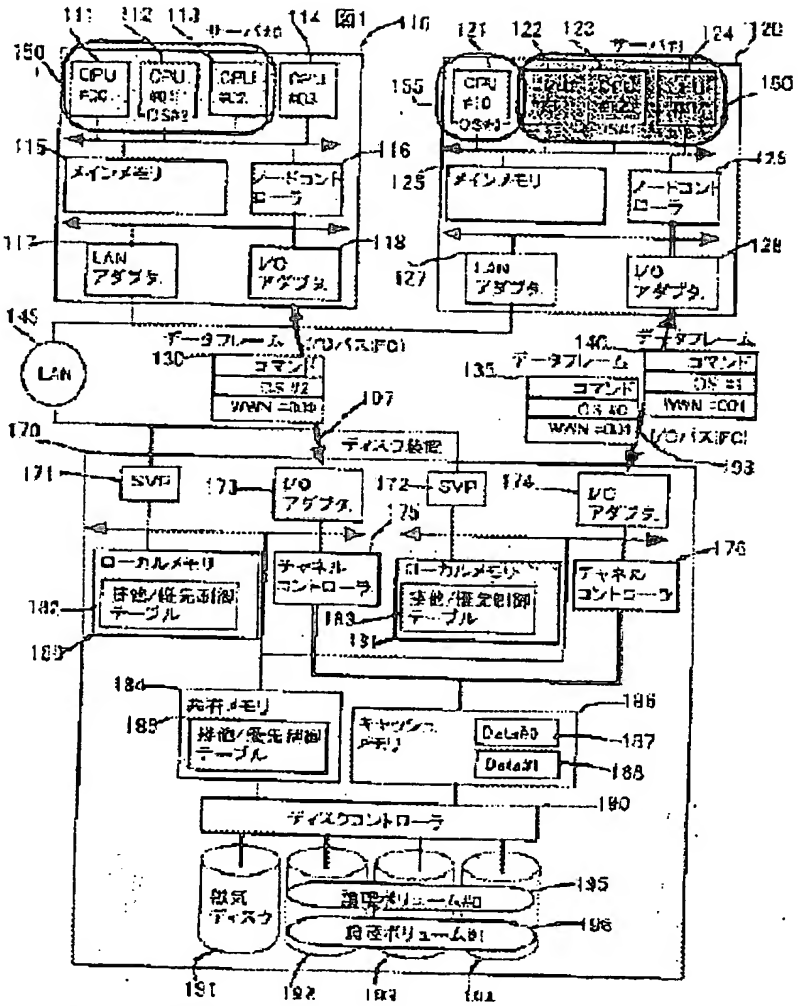
MicroPatent® PatSearch FullText: Record 1 of 1

Search scope: JP (bibliographic data only)
Years: 2001-2003
Patent/Publication No.: JP2002222110



JP2002222110 A
STORAGE SYSTEM AND VIRTUAL PRIVATE VOLUME CONTROLLING METHOD
HITACHI LTD
Inventor(s):TANAKA ATSUSHI ;OBARA KIYOHIRO ;ODAWARA HIROAKI
Application No. 2001016503 JP2001016503 JP, Filed 20010125,A1 Published 20020809Published 20020809

Abstract: PROBLEM TO BE SOLVED: To change an exclusive control method and a priority control method for improving security ability and optimizing performance of a system even when there are a plurality of operating systems in a server and a path to the disk device is shared between them by making discrimination between a plurality of operating systems and applications on the disc device side.

SOLUTION: In this virtual private volume controlling device, an identification number for identifying the operating system is given to the inside of a command issued by the operating system, and according to the identification number, exclusion and priority of access is controlled inside the disk device. In this way, transaction performance of the server and the disk device as a whole can be improved.

Int'l Class: G06F01200; G06F015177

Patents Citing this One: No US, EP, or WO patents/search reports have cited this patent. MicroPatent Reference Number: 000581093

Home Search List

For further information, please contact:
Technical Support | Billing | Sales | General Information

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－222110
（P2002－222110A）
(43)公開日　平成14年８月９日(2002.8.9)

(51)Int.Cl.⁷		識別記号	FI		テーマコード（参考）
Ｇ０６Ｆ	12/00	５３７	Ｇ０６Ｆ	12/00	５３７Ａ　５Ｂ０４５
		５０１			５０１Ａ　５Ｂ０８２
	15/177	６８２		15/177	６８２Ｇ

審査請求　未請求　請求項の数24　ＯＬ　（全 24 頁）

(21)出願番号　特願2001－16503(P2001－16503)

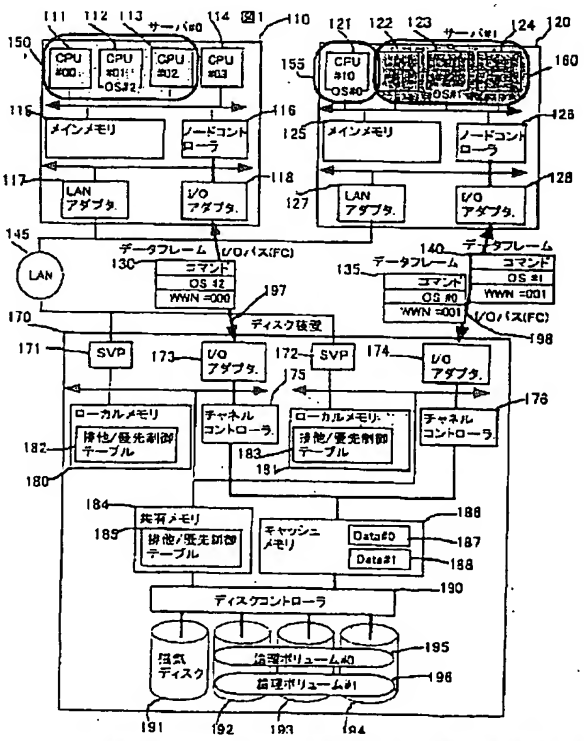
(22)出願日　平成13年１月25日(2001.1.25)

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　田中　淳
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内
(72)発明者　小原　清弘
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内
(74)代理人　100075096
弁理士　作田　康夫

最終頁に続く

(54)【発明の名称】　ストレージシステム及び仮想プライベートボリューム制御方法

(57)【要約】
【課題】　サーバ内に複数のＯＳが存在し、ディスク装置へのパスを共有していると、ディスク装置側ではこれら複数のＯＳ、アプリケーションを区別することができない。そこで、複数のＯＳ、アプリケーションを区別して、排他制御、優先制御方法を代えることでセキュリティ能力を向上させ、システムの性能最適化を図る。
【解決手段】　ＯＳが発行するコマンドの内部にＯＳを識別する識別番号を付与し、ディスク装置内でその識別番号を元にアクセスの排他、優先処理を制御することで、サーバ及びディスク装置全体のトランザクション性能を高めることを特徴とする仮想プライベートボリューム制御装置とする。



【特許請求の範囲】

【請求項１】複数のＯＳを識別する識別番号を付与されたコマンドを受信し、前記識別番号を抽出し、前記コマンドに対応した論理ボリュームへのアクセスを処理するかまたはアクセスを拒否するかに相当するレスポンスを前記識別番号に付与して返信することを特徴とするストレージシステム。

【請求項２】前記論理ボリュームは複数の磁気ディスク装置から構成されることを特徴とする請求項１記載のストレージシステム。

【請求項３】前記受信したコマンドの識別番号に対応してアクセス処理の優先順位を変えることを特徴とする請求項１または２記載のストレージシステム。

【請求項４】前記受信したコマンドのアクセスを処理するかまたはアクセスを拒否するかをあらかじめ設定した条件に基づき判断し、前記レスポンスを返信することを特徴とする請求項１乃至３記載のストレージシステム。

【請求項５】前記識別番号と前記コマンドのアクセスの処理または拒否との関係を記憶したテーブルを有することを特徴とする請求項項１乃至４記載のストレージシステム。

【請求項６】前記受信したコマンドの複数の識別番号の組み合わせに対応してレスポンスを返信することを特徴とする請求項１乃至５記載のストレージシステム。

【請求項７】複数のＯＳを有し、前記ＯＳからのアクセス要求があると当該ＯＳを識別する識別番号を付与し、前記付与された識別番号を有するコマンドを送信するサーバと、当該送信されたコマンドを受信するディスク装置とを備え、前記ディスク装置は前記識別番号を抽出し、前記抽出した識別番号に対応した論理ボリュームへのアクセスを処理するかまたはアクセスを拒否するかに相当するレスポンスを前記識別番号を付与して返信し、前記サーバは前記レスポンスを受信することを特徴とする仮想プライベートボリューム制御方法。

【請求項８】前記サーバは前記ＯＳを識別する識別番号をデータフレームに書き込み、当該データフレームをコマンドとして送信し、前記ディスク装置は当該データフレームを受信して前記識別番号を抽出することを特徴とする請求項７記載の仮想プライベートボリューム制御方法。

【請求項９】前記受信したコマンドの識別番号に対応してアクセス処理の優先順位を変えることを特徴とする請求項７または８記載の仮想プライベートボリューム制御方法。

【請求項１０】前記受信したコマンドのアクセスを処理するかまたはアクセスを拒否するかをあらかじめ設定した条件に基づき判断し、前記レスポンスを返信することを特徴とする請求項７乃至８記載の仮想プライベートボリューム制御方法。

【請求項１１】前記識別番号と前記コマンドのアクセスの処理または拒否との関係を記憶したテーブルを有することを特徴とする請求項７乃至１０記載の仮想プライベートボリューム制御方法。

【請求項１２】複数ＯＳの少なくとも一つからアクセス要求があると当該ＯＳを識別する識別番号を付与し、前記付与された識別番号をサーバ内に記憶し、前記識別番号に対応するレスポンスを受信し、当該レスポンスを前記ＯＳへ返信することを特徴とするＯＳ管理ソフトプログラム。

【請求項１３】複数のＯＳおよびアプリケーションを識別する識別番号を付与されたコマンドを受信し、前記識別番号を抽出し、前記コマンドに対応した論理ボリュームへのアクセスを処理するかまたはアクセスを拒否するかに相当するレスポンスを前記識別番号に付与して返信することを特徴とするストレージシステム。

【請求項１４】前記論理ボリュームは複数の磁気ディスク装置から構成されることを特徴とする請求項１３記載のストレージシステム。

【請求項１５】前記受信したコマンドの識別番号に対応してアクセス処理の優先順位を変えることを特徴とする請求項１３または１４記載のストレージシステム。

【請求項１６】前記受信したコマンドのアクセスを処理するかまたはアクセスを拒否するかをあらかじめ設定した条件に基づき判断し、前記レスポンスを返信することを特徴とする請求項１３乃至１５記載のストレージシステム。

【請求項１７】前記識別番号と前記コマンドのアクセスの処理または拒否との関係を記憶したテーブルを有することを特徴とする請求項項１３乃至１６記載のストレージシステム。

【請求項１８】前記受信したコマンドの複数の識別番号の組み合わせに対応してレスポンスを返信することを特徴とする請求項１乃至１７記載のストレージシステム。

【請求項１９】複数のアプリケーションおよびＯＳを有し、前記アプリケーションおよびＯＳからのアクセス要求があると当該アプリケーションおよびＯＳを識別する識別番号を付与し、前記付与された識別番号を有するコマンドを送信するサーバと、当該送信されたコマンドを受信するディスク装置とを備え、前記ディスク装置は前記識別番号を抽出し、前記抽出した識別番号に対応した論理ボリュームへのアクセスを処理するかまたはアクセスを拒否するかに相当するレスポンスを前記識別番号を付与して返信し、前記サーバは前記レスポンスを受信することを特徴とする仮想プライベートボリューム制御方法。

【請求項２０】前記サーバは前記アプリケーションおよびＯＳを識別する識別番号をデータフレームに書き込み、当該データフレームをコマンドとして送信し、前記ディスク装置は当該データフレームを受信して前記識別番号を抽出することを特徴とする請求項１９記載の仮想

プライベートボリューム制御方法。

【請求項２１】前記受信したコマンドの識別番号に対応してアクセス処理の優先順位を変えることを特徴とする請求項１９または２０記載の仮想プライベートボリューム制御方法。

【請求項２２】前記受信したコマンドのアクセスを処理するかまたはアクセスを拒否するかをあらかじめ設定した条件に基づき判断し、前記レスポンスを返信することを特徴とする請求項１９乃至２１記載の仮想プライベートボリューム制御方法。

【請求項２３】前記識別番号と前記コマンドのアクセスの処理または拒否との関係を記憶したテーブルを有することを特徴とする請求項１９乃至２２記載の仮想プライベートボリューム制御方法。

【請求項２４】複数アプリケーションの少なくとも一つからアクセス要求があると当該アプリケーションおよびＯＳを識別する識別番号を付与し、前記付与された識別番号をサーバ内に記憶し、前記識別番号に対応するレスポンスを受信し、当該レスポンスを前記アプリケーションおよびＯＳへ返信することを特徴とするアプリケーションおよびＯＳ管理ソフトプログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、サーバおよび記憶装置技術に関し、特に複数の記憶装置、ＣＰＵなどの構成要素を接続し、高いセキュリティ能力、高い応答性能、トランザクション数が要求される情報処理装置、記憶装置に有効な技術に関する。

【０００２】

【従来の技術】複数のＣＰＵ、メモリとＩ／Ｏアダプタを持つサーバと大容量のディスク装置を組み合わせ、複数のユーザでそれら資源を共用する環境が現実になってくるに従って、各ユーザのセキュリティを守り、効率的に資源を使うことが要求されてきている。従来のサーバ及びディスク装置においては、論理ボリュームの排他制御を行うためにサーバに内蔵しているＩ／Ｏアダプタ毎に識別番号を持ちそれをストレージ側で区別する方式が採用されている。図２に従来のサーバ、ディスク装置構成例を示す。ここでサーバ＃０（１１０）とサーバ＃１（１２０）はそれぞれ複数のＯＳ＃０（１５５）～ＯＳ＃２（１２０）を内蔵している。またＩ／Ｏアダプタ１１８、１２８を持ちストレージ１７０に接続している。ストレージ１７０にアクセスする場合データフレーム１３０、１３５、１４０をＩ／Ｏアダプタ１１８、１２８より発行する。データフレーム１３０、１３５、１４０の記述内容の中にはＷｏｒｌｄ Ｗｉｄｅ Ｎａｍｅ（ＷＷＮ）（２０５、２１５、２２５）がありＩ／Ｏアダプタ１１８、１２８毎に固有の認識番号をそこに書きこむ。一方ディスク装置１７０ではＩ／Ｏアダプタ１７３、１７４を持ち上記データフレーム１３０、１３５、１４０を受信する。ディスク装置１７０で内部制御を行うチャネルコントローラ１７５、１７６は受信したデータフレーム１３０、１３５、１４０から識別番号であるＷＷＮ２０５、２１５、２２５を抽出する。そしてチャネルコントローラ１７５、１７６はＷＷＮ２０５、２１５、２２５とローカルメモリ１８０、１８１にあらかじめ登録されている排他／優先制御テーブル１８２、１８３を比較することで論理ボリューム１９５、１９６に対するアクセスの排他／優先制御を行う。例えばサーバ＃０（１１０）のＯＳ＃２（１５０）が論理ボリューム＃０（１９５）を所有し、サーバ＃１（１２０）のＯＳ＃１（１６０）が論理ボリューム＃１（１９６）を所有していることを仮定する。ここで誤ってＯＳ＃１（１６０）が論理ボリューム＃０（１９５）をアクセスした場合、このアクセスのデータフレーム１４０のＷＷＮ２２５には識別番号００１があるので、チャネルコントローラ１７６は排他／優先制御テーブル１８３を参照し、識別番号０００しかアクセスが許可されていないことがわかる。そこでチャネルコントローラ１７６はＯＳ＃１（１６０）に対してアクセスが拒否されたことを伝える。このような論理ボリュームの排他／優先制御を備えたディスク装置の技術については、特開２０００-２０４４７号に開示された「記憶サブシステム」等がある。

【０００３】一方サーバ内に複数の資源が存在する場合それらを複数のＯＳで分割して使う技術が存在する。この技術を用いることで、サーバのＣＰＵ、メモリ等を効率よく利用することが可能になってきている。これらの技術の一例としてＬＰＡＲ（Ｌｏｇｉｃａｌ ＰＡＲＴｉｏｎｉｎｇ）がある。これを用いた場合、複数のＯＳを管理するソフトウエアプログラムとしてハイパバイザ４３０が存在することにより、動的にＯＳに，ＣＰＵ、メモリ、Ｉ／Ｏデバイスを割り当てることが可能となる。例えば図２のサーバ＃１（１２０）では、ＣＰＵ＃１０～＃１３と４個あるので、ＬＰＡＲを用いることでＯＳ＃０（１５５）にはＣＰＵ＃１０を割り当て、ＯＳ＃１（１６０）にはＣＰＵ＃１１、＃１２、＃１３（１２２、１２３、１２４）を割り当てることができる。メインメモリ１２５についても容量を適当な割合でＯＳ＃０（１５５）、ＯＳ＃１（１６０）に割り当てることができる。Ｉ／Ｏアダプタが複数ある場合はＯＳ＃０（１５５）、ＯＳ＃１（１６０）に割り当てることができるがこの例では１個しかないので、Ｉ／Ｏアダプタ１２８は両ＯＳで共有することになる。このような技術については、特開平１０-３０１７９５号に開示された「仮想計算機システム」等がある。また、特開２０００－１１２８０４号には、複数のＯＳがファイルを共用して動作する仮想計算機システムにおいて、１つのＯＳが他のＯＳのデータファイルをアクセスすることを防止するため、ＯＳ番号が付されたＩ／Ｏ命令をディスク装置に対して発行することが記載されている。また、特開平４－

１４０８４４号には、ＯＳが発行したＩ／Ｏ命令を解析するＩ／Ｏ解析手段を有し、Ｉ／Ｏ解析手段では引き取ったＩ／Ｏ命令にＯＳ識別子を付加し、複数ＯＳを同時動作させたときに排他制御することが記載されている。

【０００４】

【発明が解決しようとする課題】以上に述べたように，ＷＷＮ２０５、２１５、２２５をディスク装置１７０で識別することによって論理ボリューム１９５、１９６の排他／優先制御を行うことが可能となる。またＬＰＡＲを用いることでサーバ内複数のＯＳに対して，ＣＰＵ、メモリ、Ｉ／Ｏデバイスを効率よく割り当てることが可能になる。しかし一般にサーバ内ＣＰＵ数の増加の割合に対して、Ｉ／Ｏデバイスの数はそれほど多く増加せず、ＣＰＵ間で共有する場合が多い。これはコストを削減とＩ／Ｏデバイスが外部へのＩ／Ｆを持つことによる物理的な大きさのためである。このような状況で論理ボリュームの排他／優先制御を行う場合、以下のように排他／優先制御が出来なくなる問題がある。例えば、図２のサーバ＃１（１２０）ではＯＳ＃０（１５５）とＯＳ＃１（１６０）が同じサーバ内に存在し、Ｉ／Ｏアダプタ１２８を共有して使うことになる。しかし両ＯＳから発行するデータフレーム１３５、１４０のＷＷＮ２１５、２２５はＩ／Ｏアダプタ１２８を共有するため同じ識別番号“００１”が付けられる。よってディスク装置１７０ではこの２個のデータフレームについて排他／優先制御は行うことができない。よってＯＳ＃０（１５５）が論理ボリューム＃０（１９５）を所有、他のＯＳにアクセスすることを許可しない場合でも、上記理由によりＯＳ＃１（１６０）は論理ボリューム＃０（１９５）にアクセスすることが可能となってしまう。本発明の主たる目的は、サーバ内の複数ＯＳがＩ／Ｏデバイスとストレージシステムを共有する場合に、各ＯＳに固有な識別情報をサーバ側で付加、ディスク装置側で抽出する手段をつけることにより各ＯＳ毎に所有する論理ボリュームの排他／優先制御を行う技術を改良することにある。本発明のさらに別の目的は、ディスク装置に排他／優先制御方法を記述したテーブルを設置し、複数の排他／優先制御方法から選択し実行する技術を提供することにある。本発明のさらに別の目的は、サーバからディスク装置に対して、各ＯＳ毎に複数の排他／優先制御方式をディスク装置に対して命令する技術を提供することにある。

【０００５】

【課題を解決するための手段】上記問題を解決するために、複数のＯＳを有し、前記ＯＳからのアクセス要求があると当該ＯＳを識別する識別番号を付与し、前記付与された識別番号を有するコマンドを送信するサーバと、当該送信されたコマンドを受信するディスク装置とを備え、前記ディスク装置は前記識別番号を抽出し、前記抽出した識別番号に対応した論理ボリュームへのアクセスを処理するかまたはアクセスを拒否するかに相当するレスポンスを前記識別番号を付与して返信し、前記サーバは前記レスポンスを受信する仮想プライベートボリューム制御方法とする。また、サーバにＯＳ毎の識別番号をディスク装置アクセス時に発行するデータフレーム内に書き込む手段をおく。また、ディスク装置の中で上記ＯＳ毎の識別番号を抽出し、それを元にアクセスの排他／優先制御を行う手段をおく。さらにサーバからディスク装置にＯＳ毎にアクセス可能な論理ボリュームまたは排他／優先制御情報を送る手段をおく。さらにディスク装置に上記ＯＳ毎にアクセス可能な論理ボリュームまたは排他／優先制御情報を格納するテーブルをおく。ここで、複数のＯＳとは、ＯＳユーザーが異なる場合などの、仮想的に異なる複数のＯＳであればよく、同種のＯＳ（例えばＯＳ＃０がWindowsＮＴ［米国およびその他の国における米国Microsoft Corp.の登録商標；以下同じ］でＯＳ＃１がWindowsＮＴの場合、ＯＳ＃１がＵＮＩＸ［X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標；以下同じ］でＯＳ＃２がＵＮＩＸなど）であってもよい。

【０００６】

【発明の実施の形態】（実施例１）以下、本発明に係るの実施例１を図面に示しさらに詳細に説明する。図１は本発明の実施例１における仮想プライベートボリューム制御装置の全体構成図である。１１０、１２０はサーバであり、ＯＳ、アプリケーションを動かし、ディスク装置１７０に対してデータの読み出し、書き込み等の命令を発行する。１１１、１１２、１１３、１１４、１２１、１２２、１２３、１２４はCPUであり、サーバ＃０、＃１（１１０、１２０）に複数格納され、OS,アプリケーションの処理を並列に行う。１１５、１２５はメインメモリであり、CPUが直接使うデータを格納してある。１１６、１２６はノードコントローラであり、CPUとI/Oアダプタまたはlanアダプタ等の下位アダプタ間の転送を制御する。１１７、１２７はLANコントローラであり、CPU（１１１、１１２、１１３、１１４、１２１、１２２、１２３、１２４）とLAN１４５に接続している外部装置間の転送を制御する。１１８、１２８、１７３、１７４はI／Oアダプタであり、CPU（１１１、１１２、１１３、１１４、１２１、１２２、１２３、１２４）とディスク装置１７０間の転送を制御する。図１の例では複数のCPUでメインメモリ、LANアダプタ、I/Oアダプタを共用することになる。１４５はLAN（Local Area Network）でありサーバとディスク装置間のデータを通す通信路である。このLAN１４５を使ってディスク装置１７０はサーバ１１０、１２０からの排他／優先制御に関する情報を入手する。１７０はディスク装置であり、サーバ＃０、＃１（１１０、１２０）の命令に従ってデータの読み出し、書き込みを行う。１７１、１７２はSVP（Service Processor）であり、LAN１４５を経由

してサーバ＃０、＃１（１１０、１２０）より送られるディスク装置１７０の制御情報を受信し、ディスク装置１７０の必要な制御情報を変更する。１７５、１７６はチャネルコントローラであり、サーバ＃０、＃１（１１０、１２０）からくるコマンドを解析し、必要なデータの検索、送受信を行う。１８０、１８１はローカルメモリであり、チャネルコントローラ１７５、１７６を制御する際に必要な制御情報を格納する。１８４は共有メモリであり、チャネルコントローラ１７５、１７６間の通信、制御を行う際に必要な制御情報を格納する。１８２、１８３、１８５は排他／優先制御テーブルであり、論理ボリューム等の排他／優先制御を行う際に必要な制御情報を格納する。詳しくは図６にて説明する。１８６はキャッシュメモリであり、サーバ＃０、＃１（１１０、１２０）が必要としているデータを格納するメモリである。ただしすべてのデータは格納されておらず、使用頻度の高いデータのみ格納されている。１８７、１８８はキャッシュメモリ１８６内に格納されたデータである。１９０はディスクコントローラであり、磁気ディスク１９１、１９２、１９３，１９４とキャッシュメモリ１８６間データの送受信制御を行う。１９１、１９２、１９３，１９４は磁気ディスクであり、データが最終的に格納されている記憶装置である。１５０、１５５、１６０はOSであり、アプリケーションからのアクセスの制御等を行う。１９５、１９６は論理ボリュームであり、OS＃０（１５０）、OS＃１（１５５）、OS＃２（１６０）に割り当てられる記憶領域である。OS＃０（１５０）、OS＃１（１５５）、OS＃２（１６０）から論理ボリューム＃０（１９５）、＃１（１９６）に対するアクセス命令はI／Oアダプタ１１８、１２８でコマンドおよび制御情報に変換され、データフレーム１３０、１３５、１４０内に書き込まれ、ディスク装置１７０へ転送される。データフレーム内の詳細については図３にて説明する。１９７、１９８はI/Oパスであり、サーバ＃０、＃１（１１０、１２０）とディスク装置１７０を接続しているデータの通信路である。この例ではファイバーチャネル（FC）を示しているが他の規格でも本発明の範囲内である。ここで、複数のＯＳとは、ＯＳユーザーが異なる場合などの、仮想的に異なる複数のＯＳであればよく、同種のＯＳ（例えばＯＳ＃０がＮＴでＯＳ＃１がＮＴの場合、ＯＳ＃１がＵＮＩＸでＯＳ＃２がＵＮＩＸなど）であってもよい。

【０００７】図３に本発明の実施例１におけるデータフレーム構造例を示す。この構造例はファイバーチャネルの標準規格のフォーマットを参考にしたものである。３００はデータフレームの例であり、幾つかの部分から成り立つデータである。３０５はStart Of Frame識別子であり、データフレームの先頭をI／Oアダプタ１１８、１２８、１７３、１７４に伝える。３１０はフレームヘッダであり、データフレームのアドレス等を記述している。詳細は後で述べる。３１５は、ペイロードであり上位プロトコルのデータ、コマンド等が格納されている。詳細は後で述べる。３２０はCRC（Cyclic Redundancy Check）であり、データフレーム３００内のデータの誤りを発見し訂正を行う。３２５はEnd Of Frame識別子であり、データフレームの終わりをI／Oアダプタ１１８、１２８、１７３、１７４に伝える。次にフレームヘッダ３１０の構造例を示す。３３０は送信先ポートIDでありデータフレーム３００が次に進みたい装置のポートIDを示す。３３５は送信元ポートIDでありデータフレームをディスク装置１７０に送信してきた装置のポートIDを示す。具体的にはWWNの情報がここに格納され、異なるＩ／Ｏアダプタからのデータフレームの排他／優先制御に用いられる。３４０は上位レイヤプロトコルであり、送信元と送信先がデータのやり取りを行う際に使うプロトコルの種類を示しており、たとえばSCSI等があげられる。３４５は上位レイヤシーケンスIDであり、上位プロトコルに従った通信におけるシーケンスの順番を示す。３５０はシーケンス内フレーム番号であり、シーケンス内にある複数のデータフレームの順番を示す。３５５は上位レイヤプロトコル送信先IDであり、最終的にデータフレーム３００が到着する装置のポートIDを示す。次にペイロード内容としてＳＣＳＩを上位プロトコルに使った場合のリードコマンド構造例を示す。３６０はグループコードであり、コマンドの長さまたは標準のコマンドかベンダユニークなコマンドかの区別をする。３６５はコマンドコードでありこの例の場合はリードコマンドに対応するコードが入る。３７０は論理ボリューム番号（ＬＵＮ）であり例えば磁気ディスク１９１～１９４に設定された　論理ボリューム＃０（１９５）、＃１（１９６）を示している。３７５はデータ等が存在する場所のアドレスであり、例えば論理ボリューム＃０（１９５）、＃１（１９６）内のアドレスを示している。３８０はＬＰＡＲ－ＩＤを示しており、具体的にはハイパバイザ４３０の指定した各ＯＳの識別番号が入る。３８５は転送データ長であり、リードする範囲を示す。３９０はコントロールバイトであり、データフレーム３００またはペイロード３１５が扱える容量に合わせるために付け加える空白なデータである。なお上記例以外のフォーマットを使ってWWN、ＬＰＡＲ－ＩＤをフレームに内蔵させた場合でも、本発明の範囲であることは自明である。図４に本発明の実施例１におけるソフトウエア構造例とデータ転送成功例を示す。ここでサーバ＃０（１１０）のみ説明する。４１０、４１１、４１２、４１３はミドル／アプリであり、ユーザの処理を直接受けるプログラムである。４２０、４２１、４２２、４２３はＯＳであり、ミドル／アプリ４１０、４１１、４１２、４１３からの要求を調整、処理を行う。４３０はハイパバイザであり、サーバ＃０（１１０）内のＣＰＵ、メインメモリ、Ｉ／Ｏデバイスを複数のＯＳ４２０、４２１、４

２２、４２３に分割して利用させる。また各ＯＳ４２０、４２１、４２２、４２３の識別番号となるＬＰＡＲ－ＩＤ４５０、４５１、４５２を管理、設定する。４４０、４４１、４４２はデバイスドライバであり、ディスク装置１７０へのアクセスが必要な場合、コマンド、データ、WWN、ＬＰＡＲ－ＩＤ等をデータフレーム１３０に設定してディスク装置１７０に送信する。４５５は制御フレームであり、サーバ＃０（１１０）からディスク装置１７０の制御方式等を変更したい場合にこの中に設定してディスク装置１７０にあるテーブルセットプログラム４７０に送信する。制御フレームの詳細については図７にて説明する。４６０、４６５はディスク装置１７０のμプログラム（μＰ）であり、ディスク装置１７０のＩ／Ｏアダプタ１７３、１７４、チャネルコントローラ１７５、１７６、キャッシュメモリ１８６を制御する。ディスク装置１７０のμプログラム４６０は、Ｉ／Ｏアダプタでフレームの受信、解析を制御するＩ／Ｏアダプタμ Ｐと、チャネルコントローラでコマンドの解析、排他／優先制御、コマンド実行を制御するチャネルコントローラμＰ（４６２）に分かれている。さらにチャネルコントローラμＰ（４６２）ではデータフレーム１３０内に書きこまれている、WWN、ＬＰＡＲ－ＩＤを抽出して、排他／優先制御テーブル１８２、１８５の内容と比較することにより、Ｉ／Ｏアクセスの排他／優先制御を実行する。排他／優先制御の詳細は図１２、１２、１３、１４で説明する。４７０はテーブルセットプログラムであり、サーバ＃０（１１０）から来た制御フレーム４５５を受信し、内容をローカルメモリ１８０または共有メモリ１８４内の排他／優先制御テーブル１８２、１８５に格納する。図４のようにデータフレーム１３０のＬＰＡＲ－ＩＤがローカルメモリ１８０、または共有メモリ１８４にある排他／制御テーブル１８２、１８５内で論理ボリューム＃０（１９５）へのアクセスを許可されている場合、アクセスの結果は以下のようになる。アクセスが許可されているので、論理ボリューム＃０（１９５）内の必要なデータをリードし、フレーム１３６にＬＰＡＲ－ＩＤといっしょに載せてサーバ＃０（１１０）に送る。さらにデータ転送が正しく処理したことをステータスとして、フレーム１３７にＬＰＡＲ－ＩＤ等といっしょに載せてサーバ＃０（１１０）に送る。これらのフレーム１３６、１３７を受け取ることによりサーバはデータアクセスを完了する。図４の様なソフトウエア構造においてハイパバイザ４３０がデバイスドライバ４４０、４４１、４４２にＬＰＡＲ－ＩＤを設定する方式は２通りある。一つは初期設定時にデバイスドライバ４４０、４４１に設定する方式である。ＯＳ４２０、４２１とデバイスドライバ＃０（４４０）、＃１（４４１）が一対一に対応している場合はＯＳにＬＰＡＲ－ＩＤが割り当てられた時に各デバイスドライバに来るＩ／ＯのＬＰＡＲ－ＩＤも決まっているので、初期設定時にハイパバイザ４３０よりＯＳ４２０、４２１、デバイスドライバ＃０（４４０）、＃１（４４１）がそれぞれＬＰＡＲ－ＩＤ＃０、＃１（４５０、４５１）を受け取り設定する。Ｉ／Ｏアクセス発生時にはデバイスドライバ＃０、＃１（４４０，４４１）は設定されたＬＰＡＲ－ＩＤ＃０、＃１（４５０，４５１）をデータフレーム１３０に設定する。２種類目はＩ／Ｏアクセス毎にデバイスドライバ＃２（４４２）にＬＰＡＲ－ＩＤ＃２／３（４５２）を連絡する方式である。ＯＳ４２２、４２３はデバイスドライバ＃２（４４２）を共有しているので実際にＩ／Ｏアクセスが発生するまでどちらのＯＳのＩ／Ｏアクセスなのか不明である。そのためＯＳがＩ／Ｏアクセスをデバイスドライバ＃２（４４２）に発行する際に、前もってハイパバイザより設定されたＬＰＡＲ－ＩＤ＃２／３（４５２）を同時にデバイスドライバ４４２に送信する。受信したデバイスドライバ＃２（４４２）はそのＬＰＡＲ－ＩＤ＃２／３（４５２）をデータフレーム１３０に設定する。以上のようにしてデータフレーム１３０にＬＰＡＲ－ＩＤ＃０、＃１、＃２、＃３（４５０、４５１，４５２）が設定されると、同じＩ／Ｏアダプタ１１８を共有している複数のＯＳ（４２０、４２１、４２２、４２３）からＩ／Ｏアクセスが発生した場合でも、ＬＰＡＲ－ＩＤ＃０、＃１、＃２、＃３（４５０、４５１，４５２）が異なるため、上で述べた様にストレージ側でこれを識別することで排他／優先制御を行うことが可能になる。図５に本発明の実施例１におけるソフトウエア構造例とデータ転送失敗例を示す。図５のようにデータフレーム１３０のＬＰＡＲ－ＩＤがローカルメモリ１８０、または共有メモリ１８４にある排他／制御テーブル１８２、１８５内で論理ボリューム＃０（１９５）へのアクセスを許可されていない場合、アクセスの結果は以下のようになる。アクセスが許可されていないので、データ転送が正しく処理できずエラーとなる。エラーが発生したことをステータスとして、フレーム１３８にＬＰＡＲ－ＩＤ等といっしょに載せてサーバ＃０（１１０）に送る。このフレーム１３８を受け取ることによりサーバは次に行うべきエラー処理を判断することが可能になる。一般的にはこの後サーバ側がディスク装置のエラー情報にアクセスして次の処理を決定する。図４，５で示したサーバへのフレーム１３７、１３８の返信は本発明のひとつの例であり、排他／優先制御に関する他の情報を返信することが可能なことは明白である。図６に本発明の実施例１における排他/優先制御テーブル例を示す。５００は例として論理ボリューム＃０（１９５）の排他／優先制御テーブルを示している。５０５、５１０、５１５、５２０は論理ボリューム＃０（１９５）アクセスする可能性があるWWNを示す。５２５、５３０、５３５、５４０は各WWNを持つI／0アダプタからアクセスする可能性のあるＯＳのＬＰＡＲ－ＩＤを示す。この表で“○”で示されているWW

Ｎ、ＬＰＡＲ－ＩＤは論理ボリューム＃０（１９５）にアクセスが可能である。逆に“×”の場合論理ボリューム＃０（１９５）にアクセスは排他される。この制御方式は図１３に詳細を説明する。また排他制御＃２と書いてあった場合は図１４で詳細を説明する排他制御方式に従う。また優先制御と書いてあった場合は図１５で示す優先制御方式に従う。ここで他の排他／優先制御を用いる場合でもこの排他／優先制御テーブル５００に登録しておけば実現可能であることは自明である。排他／優先制御テーブルは論理ボリューム毎に存在し、それらの集合はローカルメモリ１８０、１８１、共有メモリ１８４内に存在する（排他／優先制御テーブル１８２、１８３、１８５）。そしてチャネルコントローラ１７５、１７６はこれらの表を参照して排他／優先制御を行う。これらの排他／優先制御テーブルはＬＡＮ１４５を通して、サーバ＃０、＃１（１１０、１２０）から送られる制御フレーム４５５をテーブルセットプログラム４７０が受信し、生成、変更することによってできる。制御フレーム４５５の詳細については図７で説明する。図７に本発明の実施例１における制御フレーム構造例を示す。制御フレーム構造例についてはＩＰ、ＵＤＰのプロトコルを参考にしたものである。またメッセージ構造例についてはＳＮＭＰ（Ｓｉｍｐｌｅ　Ｎｅｔｗｏｒｋ　Ｍａｎａｇｅｍｅｎｔ　Ｐｒｏｔｏｃｏｌ）を参考にしたものである。ローカルネットワークヘッダ６０５には、ＬＡＮ通信に必要なアドレス、データ長等の制御情報が格納されている。ＩＰヘッダ６１０にはＩＰを使った通信に必要なアドレス、データ長等の制御情報が格納されている。ＵＤＰヘッダ６１５にはＵＤＰを使った通信に必要なアドレス、データ長等の制御情報が格納されている。ここで取り上げた以外のプロトコルを使っても本発明の範囲であることは自明である。６２０はメッセージであり詳細については後で説明する。６２５はローカルネットワークトレイラであり、データのエラーチェックのコード、データの終端のコード等が記述されている。次にメッセージ６２０の構造例を示す。６３０はコマンドフィールドでありこのメッセージを使ってディスク装置が行うべきことを示している。この例ではパラメータを排他／優先制御テーブルにセットする命令が入っている。６３５はデータ長フィールドであり、以下のデータフィールドが何個続くかを示している。この例では６個続くことを示している。６４０～６６５まではデータフィールドであり、排他／優先制御テーブルに設定すべきパラメータが記述されている。この例では、論理ボリューム＃２（６４０）のＬＡＰＲ－ＩＤ＃２（６４５）かつＷＷＮ＃１（６５０）のアクセスに関しては排他制御＃２（６５５）を行い、その制約条件は最大排他処理時間（６６０）と最大排他処理Ｉ／Ｏ数（６６５）であることを示している。データフィールドにはこのほかに優先制御を行う際の優先順位等が入ることも考えられる。

図８に本発明の実施例１におけるサーバVPV制御初期化のフロー示す。ここではサーバ＃０（１１０）側仮想プライベートボリューム（ＶＰＶ）制御の初期化を行う。ステップ７００からサーバＶＰＶ制御初期化が始まる。ステップ７０５ではハイパバイザ４３０の初期化をまず行う。ステップ７１０ではＬＰＡＲ－ＩＤ＃０（４５０）、＃１（４５１）、＃２／３（４５２）を各ＬＰＡＲ（ＯＳ）に割り当てる。ステップ７１５では各ＬＰＡＲ内の特定エリアにＩＤを格納する。ステップ７２０ではＯＳ４２０、４２１、４２２、４２３とデバイスドライバ４４０、４４１、４４２を初期化して立ち上げる。ステップ７２５ではＯＳ４２０、４２１、４２２、４２３間でデバイスドライバ４４０、４４１、４４２を共用しているかを調べる。もし共有していなければ（ＯＳ４２０、４２１の場合）デバイスドライバ４４０、４４１が各ＬＰＡＲが格納しているＬＰＡＲ－ＩＤ＃０（４５０）、＃１（４５１）を取得する。デバイス４４０、４４１、４４２を共有していれば（ＯＳ４２２、４２３の場合）、ＯＳ４２２、４２３がＬＰＡＲ－ＩＤ＃２／３（４５２）を取得する。ステップ７４０ではＬＰＡＲ－ＩＤ＃０（４５０）、＃１（４５１）、＃２／３（４５２）、論理ボリューム＃０、＃１（１９５，１９６）排他／優先制御情報をディスク装置１７０に対して制御フレーム４５５を使って送信する。そしてステップ７４５でサーバＶＰＶ制御初期化が終了する。図９に本発明の実施例１におけるサーバ側Ｉ／Ｏ処理（デバイスドライバ非共有）のフローチャート示す。ここでは初期化終了後通常のＩ／Ｏ処理のフローを示す。またデバイスドライバはＯＳ間で共有しない。ステップ８００からサーバ側Ｉ／Ｏ処理が開始される。ステップ８０５ではＯＳ４２０、４２１にＩ／Ｏ処理が来ているかどうかを調べる。もし来ていなければ、そのままステップ８０５に戻る。Ｉ／Ｏ処理が来ていればステップ８１０に進み、ＯＳ４２０、４２１がデバイスドライバ４４０、４４１用のシステムコールを起動する。ステップ８１５ではデバイスドライバ４４０、４４１が起動する。ステップ８２０ではデバイスドライバ４４０、４４１がデータフレーム１３０に自分のＬＰＡＲ－ＩＤ＃０（４５０）、＃１（４５１）を埋め込む。ステップ８３０ではディスク装置１７０からデータフレームに対するステータスが返送されたかどうかを調査する。返送していなければ、そのままステップ８３０に戻る。ステータスが返送された場合、ステップ８３５に進み、Ｉ／Ｏステータスが正常終了したかどうかを調べる。もし正常終了した場合はステップ８０５に戻る。異常終了した場合、ステップ８４０に進みエラー原因をディスク装置１７０に問い合わせる。ステップ８４５では問い合わせた結果を見て、サーバ＃０（１１０）のエラー処理を行う。ステップ８５０ではＩ／Ｏ処理の回復は不可能かどうか調べる。もし回復が可能ならばステップ８０５に進む。不可能ならばステップ

８５５に進みサーバ側Ｉ／Ｏ処理を終了する。図１０に本発明の実施例１におけるサーバ側I／0処理（デバイスドライバ共有）のフローチャートを示す。ここでは初期化終了後通常のＩ／Ｏ処理のフローを示す。またデバイスドライバはＯＳ間で共有する。ステップ９００からサーバ側Ｉ／Ｏ処理が開始される。ステップ９０５ではＯＳ４２２、４２３にＩ／Ｏ処理が来ているかどうかを調べる。もし来ていなければ、そのままステップ９０５に戻る。Ｉ／Ｏ処理が来ていればステップ９１０に進み、ＯＳ４２２、４２３がデバイスドライバ４４２用のシステムコールを起動する。ステップ９１５ではＯＳ４２２、４２３がデバイスドライバ４４２に対して自分のＬＰＡＲ－ＩＤ＃２／３（４５２）を連絡する。ステップ９２０ではデバイスドライバ４４２がデータフレーム１３０に連絡されたのＬＰＡＲ－ＩＤ＃２／３（４５２）を埋め込む。ステップ９３０以降は図９のステップ８３０以降と同じなので以下説明を省略する。図１１に本発明の実施例１におけるディスク装置VPV制御初期化のフローチャートを示す。ここではディスク装置１７０側仮想プライベートボリューム（ＶＰＶ）制御の初期化を行う。ステップ１０００からディスク装置ＶＰＶ制御初期化を開始する。ステップ１００５ではディスク装置１７０の初期化を行う。ステップ１０１０ではＶＰＶ制御μプログラム立ち上げ、排他／優先制御テーブル１８２、１８３、１８５の初期化を行う。ステップ１０１５ではサーバ＃０（１１０）より制御フレーム４５５を通して、ＬＰＡＲ－ＩＤ、論理ボリューム、排他／優先制御に関する情報を受信する。受信なければ、ステップ１０１５に戻る。受信した後、ステップ１０２０では排他／優先制御テーブル１８２、１８３、１８５の設定を行う。そしてステップ１０２５でディスク装置ＶＰＶ制御初期化が終了する。図１２に本発明の実施例１におけるディスク装置側I／0処理フローチャートを示す。ここでは初期化終了後ディスク装置１７０における通常のＩ／Ｏ処理のフローを示す。ステップ１１００よりディスク装置側Ｉ／Ｏ処理が開始される。ステップ１１０５ではＩ／Ｏアダプタ１７３、１７４にＩ／Ｏ処理が来ているかどうかを調べる。もし来ていなければ、そのままステップ１１０５に戻る。Ｉ／Ｏ処理が来ていればステップ１１１０に進み、Ｉ／ＯアダプタμプログラムＩ４６１を起動する。ステップ１１１５ではチャネルコントローラμプログラム４６２を起動する。ステップ１１２０ではコマンドに埋め込まれたＬＰＡＲ－ＩＤ３８０を抽出する。ステップ１１２５ではＩ／Ｏ処理対象の論理ボリューム３７０の排他／優先制御テーブル５００を参照する。ステップ１１３０ではＩ／ＯのＬＰＡＲ－ＩＤ３８０、WWN３３５を参照してそれが排他制御対象かどうか調べる。排他制御対象でなければ、ステップ１１５０の優先制御処理に進む。ステップ１１５０については図１５で詳細を説明する。排他制御対象ならばステップ１１３５に進む。ステップ１１３５については図１３、１３で詳細を説明する。ステップ１１５０の優先制御処理の後、ステップ１１３５で通常処理と判定された場合はステップ１１５５にてＩ／Ｏアクセス処理のキューに入ってキャッシュメモリ１８６または磁気ディスク１９１、１９２、１９３、１９４のＩ／Ｏ処理を待つ。ステップ１１３５で排他処理と判定された場合はステップ１１４０に進む。ステップ１１６０ではキューに入れたＩ／Ｏの処理が正常に終わったかどうかを調べる。正常終了した場合はステップ１１４５へ進む。異常終了した場合はステップ１１４０へ進む。ステップ１１４０ではエラーメッセージを作成してディスク装置１７０内に保存する。ステップ１１４５はデータまたはステータスメッセージを作成して、サーバに返答する。その後ステップ１１０５に戻る図１３に本発明の実施例１における第一の排他制御処理例のフローチャートを示す。ここでは、図１２のステップ１１３５の具体的な制御方式として、一番単純な排他制御のフローを示す。ステップ１２００より排他制御処理＃１が始まる。ステップ１２０５ではアクセスが排他されるべきかどうかを調べる。排他ならばステップ１２１５の排他処理へ進む。排他でなければステップ１２０５の通常処理へ進む。図１４に本発明の実施例１における第二の排他制御処理例のフローチャートを示す。ここでは、図１２のステップ１１３５の具体的な制御方式として、処理時間、処理Ｉ／Ｏ数を条件とした、制限付き排他制御のフローを示す。ステップ１３００より排他制御処理＃２が始まる。ステップ１３０５ではそれまでに排他処理を行った時間が、あらかじめ設定してある最大排他処理時間（６６０）より少ないかどうか調べる。少なければステップ１３１０へ進む。同じか、大きければステップ１３３５の通常処理へ進む。ステップ１３１０では排他処理時間を加算する。ステップ１３１５ではそれまでに排他処理を行ったＩ／Ｏ数が、あらかじめ設定してある最大排他処理Ｉ／Ｏ数（６６５）より少ないかどうか調べる。少なければステップ１３２０に進む。同じか、大きければステップ１３３５の通常処理へ進む。ステップ１３２０では排他処理Ｉ／Ｏ数の加算を行う。ステップ１３２５ではではアクセスが排他されるべきかどうかを調べる。排他ならばステップ１３３０の排他処理へ進む。排他でなければステップ１３３５の通常処理へ進む。図１５に本発明の実施例１における優先制御処理例のフローチャートを示す。ここでは排他制御を行わず、処理の順番を変更するフローを示す。この例はＳＣＳＩの標準規格のタグキューイングを参考にしたものである。ステップ１４００から優先制御処理が行われる。ステップ１４０５では処理するＩ／Ｏの処理順番を決めるためのポインタをキューの先頭にセットする。ステップ１４１０では処理するＩ／Ｏの優先順位を読み出す。ステップ１４１５ではキュー内のポインタの位置にＩ／Ｏがあるかどうかを調べる。なければ

ステップ１４３０へ進む。Ｉ／Ｏがあればステップ１４２０に進みポインタ位置にあるキュー内Ｉ／Ｏの優先順位を読み出す。ステップ１４２５ではキュウ内Ｉ／Ｏの優先順位が処理するＩ／Ｏの優先順位より高いかどうかを調べる。優先順位高ければステップ１４３５に進み、ポインタを１加算して、ステップ１４１５に戻る。優先順位が低ければステップ１４３０へ進む。ステップ１４３０では処理するＩ／Ｏの順番を現時点でセットされたポインタの位置に決定する。ステップ１４４０では通常処理へ進む。なお以上のフロー例以外の方式を使った排他／優先制御処理も本発明の範囲であることは自明である。以上図１から図１５までに示した実施例１によれば、ＬＰＡＲ－ＩＤをデータフレームに入れてディスク装置に送信し、受信したディスク装置側で排他制御を行うことにより、Ｉ／ＯアダプタをＯＳに関しても排他制御を行うことが可能となる。またディスク装置側でＬＰＡＲ－ＩＤ、ＷＷＮを考慮した排他／優先制御テーブルを持つことにより、サーバ間をまたがったＯＳ等に関してもディスク装置側で統一的に排他／優先制御を行うことが可能となる。また排他／優先制御方式も複数の方式を選択できるようにしているので、柔軟な排他制御、性能の最適化が可能となる。

（実施例２）以下、本発明に係るの実施例２を図面に示しさらに詳細に説明する。

【０００８】図１６に本発明の実施例２におけるソフトウエア構造例を示す。実施例１では排他／優先制御はディスク装置１７０で行ったが、単体サーバ＃０（１１０）の内部だけに制御を限定し、サーバ＃０（１１０）、ハイパバイザ１５００の処理能力が高い場合は、その他にハイパバイザ１５００行うことも可能である。この場合、すべてのＩ／Ｏアクセスはハイパバイザ１５００で受け取り、ハイパバイザ内に格納されているＬＰＡＲ－ＩＤ１５１０とその優先／排他制御テーブル１５５０を用いることで、排他／優先制御を行う。アクセスが許可された場合Ｉ／Ｏアクセスはそのままディスク装置１７０に送信され、アクセスが許可されない場合はそのままエラーでＯＳ（４２０）に返される。以上図１６に示した実施例によればディスク装置１７０、データフレーム１４０に変更を加えることなく、ＯＳ間の排他／優先制御を行うことが可能になる。

（実施例３）以下、本発明に係るの実施例３を図面に示しさらに詳細に説明する。

【０００９】図１７に本発明の実施例３におけるソフトウエア構造例を示す，実施例１、２では排他／優先制御を行う単位は、ＬＰＡＲ－ＩＤを持つOS単位で行っていた。しかしユーザによっては、アプリケーション単位で排他を行う必要がでてくる。同一OS内のアプリケーションならば、OSが排他／優先制御を行うことになるが、複数OSにあるアプリケーション間で行うためには実施例１、２と同様にディスク装置４６０またはハイパバイザ４３０で排他／優先制御を行うことになる。たとえばアプリケーション＃１（１６０５、１６１５　以下アプリ＃１と呼ぶ）のデータアクセスは優先したい場合、アプリ＃１（１６０５、１６１５）のアクセスが発生時にアプリケーション毎にあるＩＤ（ＡＰＬ－ＩＤ）をＬＰＡＲ－ＩＤと同様にデータフレーム１６３５につけ、ディスク装置４６０に送信する。ディスク装置４６０側では、前もって受信した制御フレーム１６２０で伝えられ排他／優先制御テーブル１６２５、１６３０に登録された、アプリケーション毎の排他／制御情報を元に排他／優先制御を行う。アクセス結果はデータ１６４０、ステータス１６４５によってサーバ＃０（１１０）に報告される。この結果として、ＬＰＡＲ－ＩＤ＃０（４５０）、ＬＰＡＲ－ＩＤ＃２（４５２）によってアクセスが排他されていない限り、アプリ＃１のデータアクセスはＯＳにかかわらず優先される。

【００１０】図１８に本発明の実施例３における排他／優先制御テーブル例を示す。実施例３ではアプリ毎に排他／優先制御を判定するために、図６に比較してテーブルが一種類増えることになる。１６２５－ａは排他／優先制御テーブル例（ＷＷＮ－ＬＰＡＲ関係）で図６の排他／優先制御テーブル５００と同様な形式である。１６２５－ｂは排他／優先制御テーブル例（ＷＷＮ－ＡＰＬ関係）であり、各ＷＷＮ（１７００～１７１５）に対するＡＰＬ（１７２０～１７３５）の排他／優先制御の有無を示してある。

【００１１】図１９に本発明の実施例３におけるデータフレーム構造例を示す。基本的には図３のデータフレーム構造と同じであるが、ペイロード３１５内にアプリケーションに関するＩＤとして、ＡＰＬ－ＩＤ（３９５）が付け加えられている。

【００１２】図２０に本発明の実施例３における制御フレーム例を示す。基本的には図７の制御フレーム構造と同じであるが、メッセージ６２０内のデータフィールドにアプリケーションに関するＩＤとして、ＡＰＬ－ＩＤ（６７０）が付け加えられている。以上図１７に示した実施例によれば、図１３～１４で示した排他／優先制御処理のフローチャートと同様な方式で、異なるOSに存在するアプリケーション間の排他／優先制御を行うことが可能になる。

【００１３】

【発明の効果】本発明によれば、複数OSがI／Oデバイスを共有してディスク装置をアクセスするとき、OS、アプリケーション毎に付与された認識番号をディスク装置が認識できるようになる。そのため、ディスク装置側でのOS、アプリケーション間のデータの排他制御が可能となりセキュリティを高めることができる。さらに、アクセスの優先制御も行うことが可能になり仮想プライベートボリュームシステム性能の最適化が図れる。

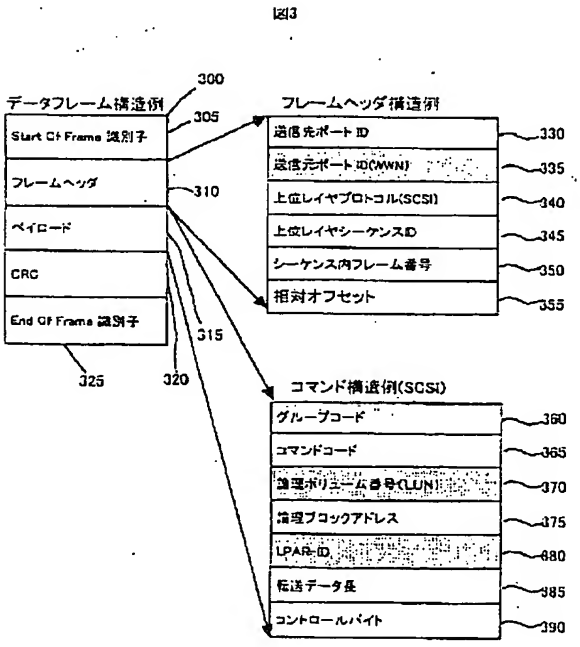
【図面の簡単な説明】

【図１】本発明の実施例１における全体構成図を示す。
【図２】従来の排他制御方式内臓記憶装置の構成図を示す。
【図３】本発明の実施例１におけるデータフレーム構造例を示す。
【図４】本発明の実施例１におけるソフトウエア構造例とデータ転送成功例を示す。
【図５】本発明の実施例１におけるソフトウエア構造例とデータ転送失敗例を示す。
【図６】本発明の実施例１における排他/優先制御テーブル例を示す。
【図７】本発明の実施例１における制御フレーム例を示す。
【図８】本発明の実施例１におけるサーバVPV制御初期化のフロー示す。
【図９】本発明の実施例１におけるサーバ側I／O処理（デバイスドライバ非共有）のフローチャート示す。
【図１０】本発明の実施例１におけるサーバ側I／O処理（デバイスドライバ共有）のフローチャートを示す。
【図１１】本発明の実施例１におけるディスク装置VPV制御初期化のフローチャートを示す。
【図１２】本発明の実施例１におけるディスク装置側I／O処理フローチャートを示す。
【図１３】本発明の実施例１における第一の排他制御処理のフローチャートを示す。
【図１４】本発明の実施例１における第二の排他制御処理のフローチャートを示す。
【図１５】本発明の実施例１における優先制御処理のフローチャートを示す。
【図１６】本発明の実施例２におけるソフトウエア構造例を示す。
【図１７】本発明の実施例３におけるソフトウエア構造例を示す。
【図１８】本発明の実施例３における排他/優先制御テーブル例を示す。
【図１９】本発明の実施例３におけるデータフレーム構造例を示す。
【図２０】本発明の実施例３における制御フレーム例を示す。

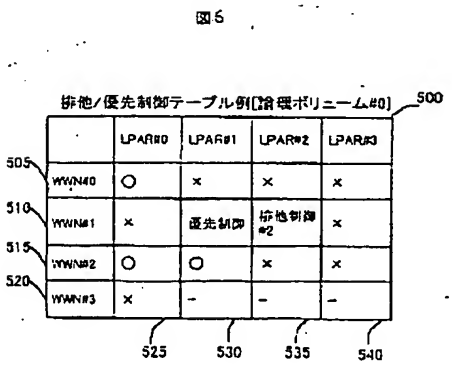
【符号の説明】

１１０、１２０・・サーバ、１１１、１１２、１１３、１１４、１２１、１２２、１２３、１２４・・ＣＰＵ、１１５、１２５・・メインメモリ、１１６、１２６・・ノードコントローラ、１１７、１２７・・ＬＡＮアダプタ、１１８、１２８、１７３、１７４・・Ｉ／Ｏアダプタ、１３０、１３５、１４０・・フレーム、１４５・・ＬＡＮ、１５０、１５５、１６０・・ＯＳ、１７０・・ディスク装置、１７１、１７２・・ＳＶＰ、１７５、１７６・・チャネルコントローラ、１８０、１８１・・ローカルメモリ、１８２、１８３、１８５・・排他／優先制御テーブル、１８４・・共有メモリ、１８６・・キャッシュメモリ、１８７、１８８・・Ｄａｔａ、１９０・・ディスクコントローラ、１９１、１９２、１９３、１９４・・磁気ディスク、１９５、１９６・・論理ボリューム、１９７、１９８・・Ｉ／Ｏパス。

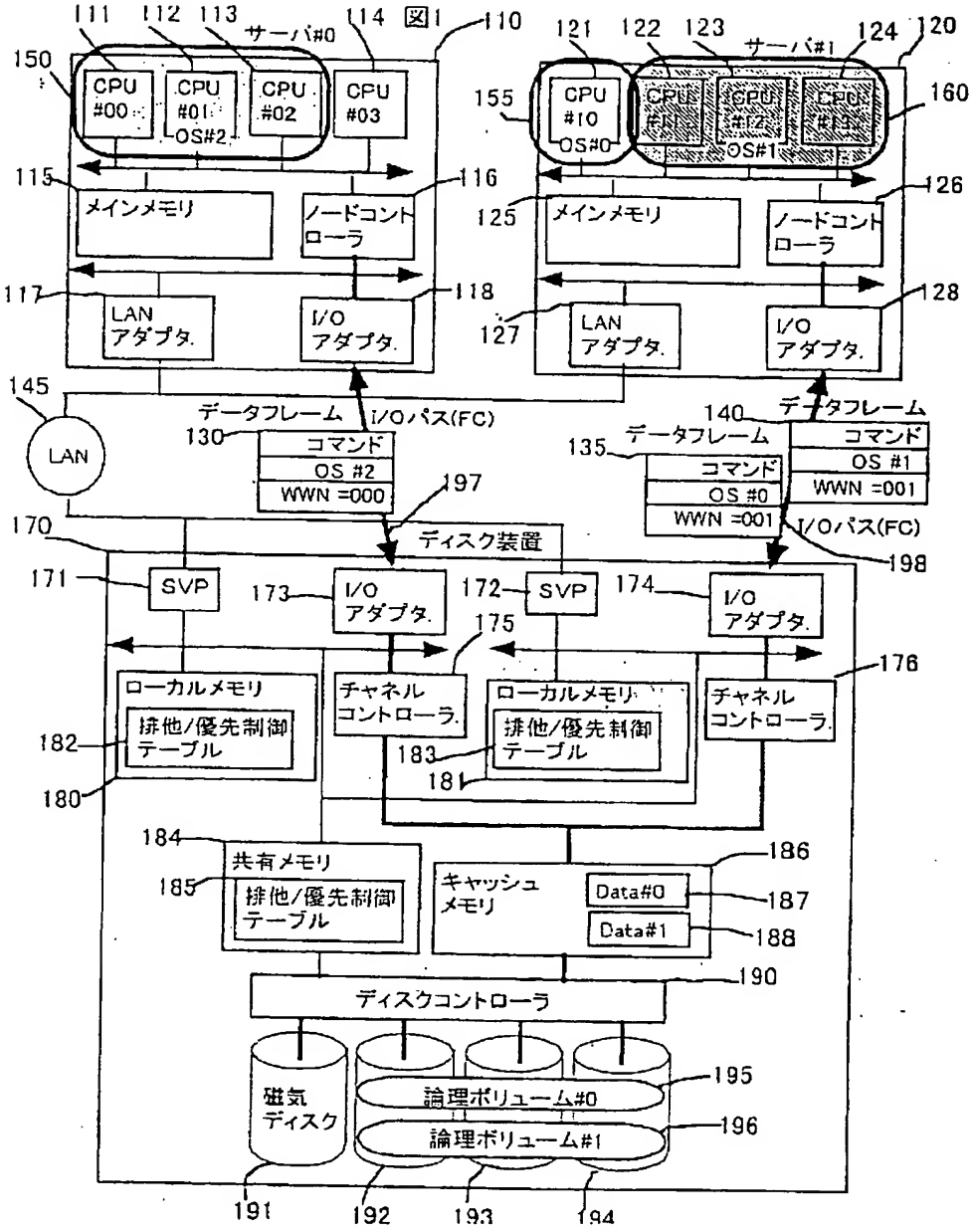
【図３】



【図６】

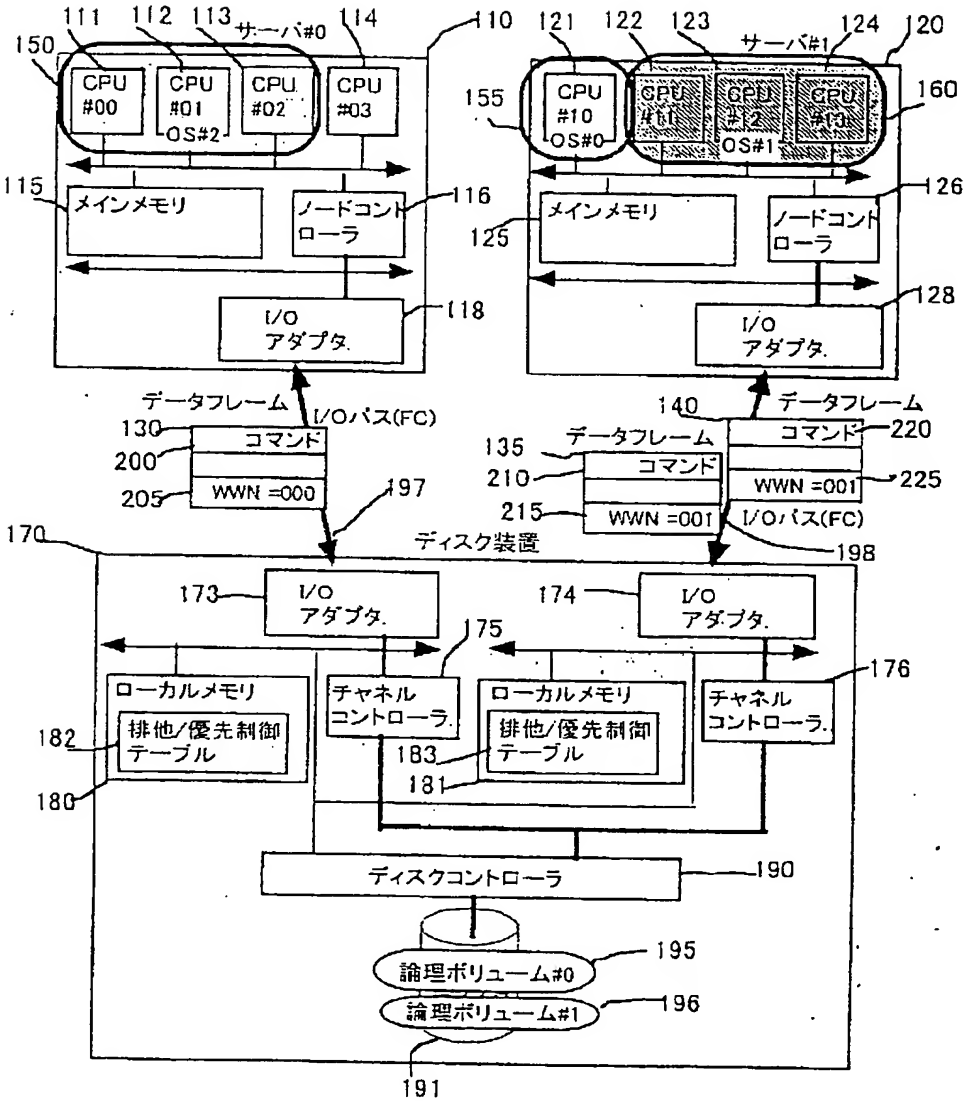


【図１】



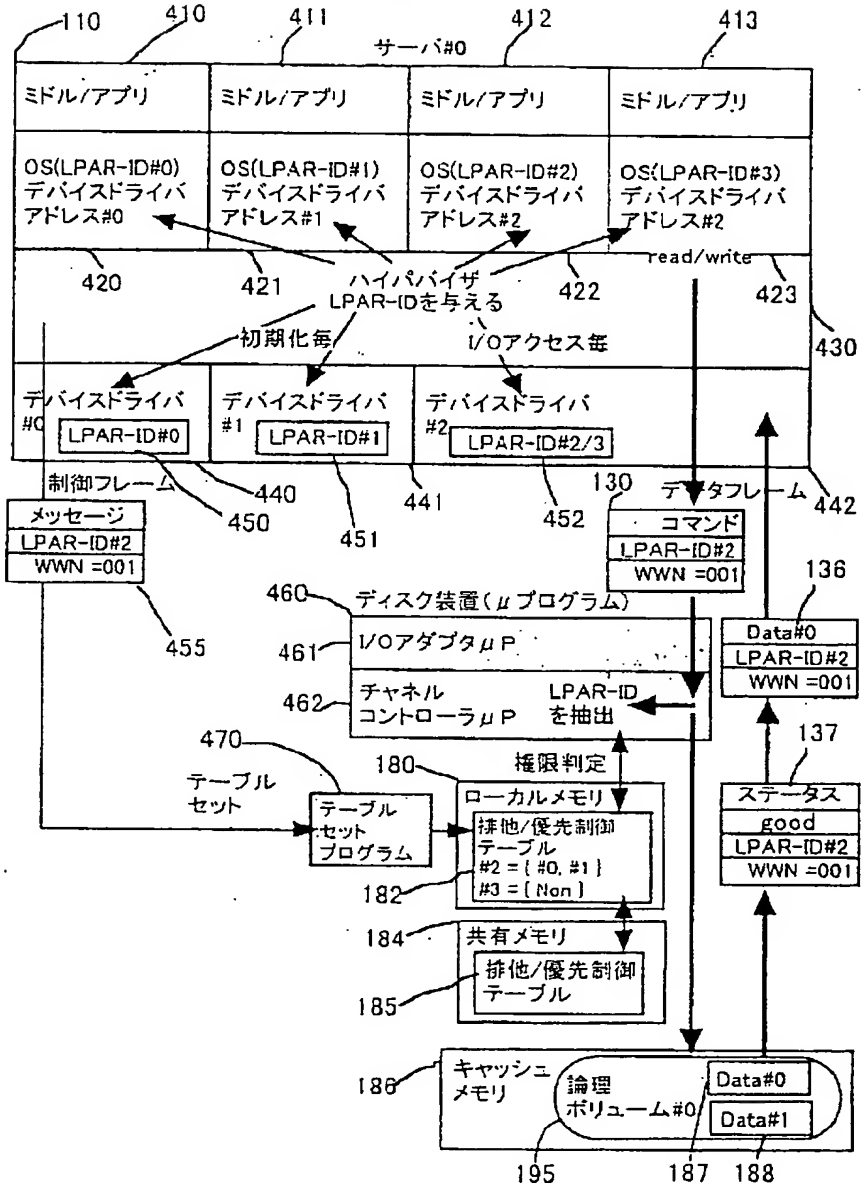
【図２】

図2



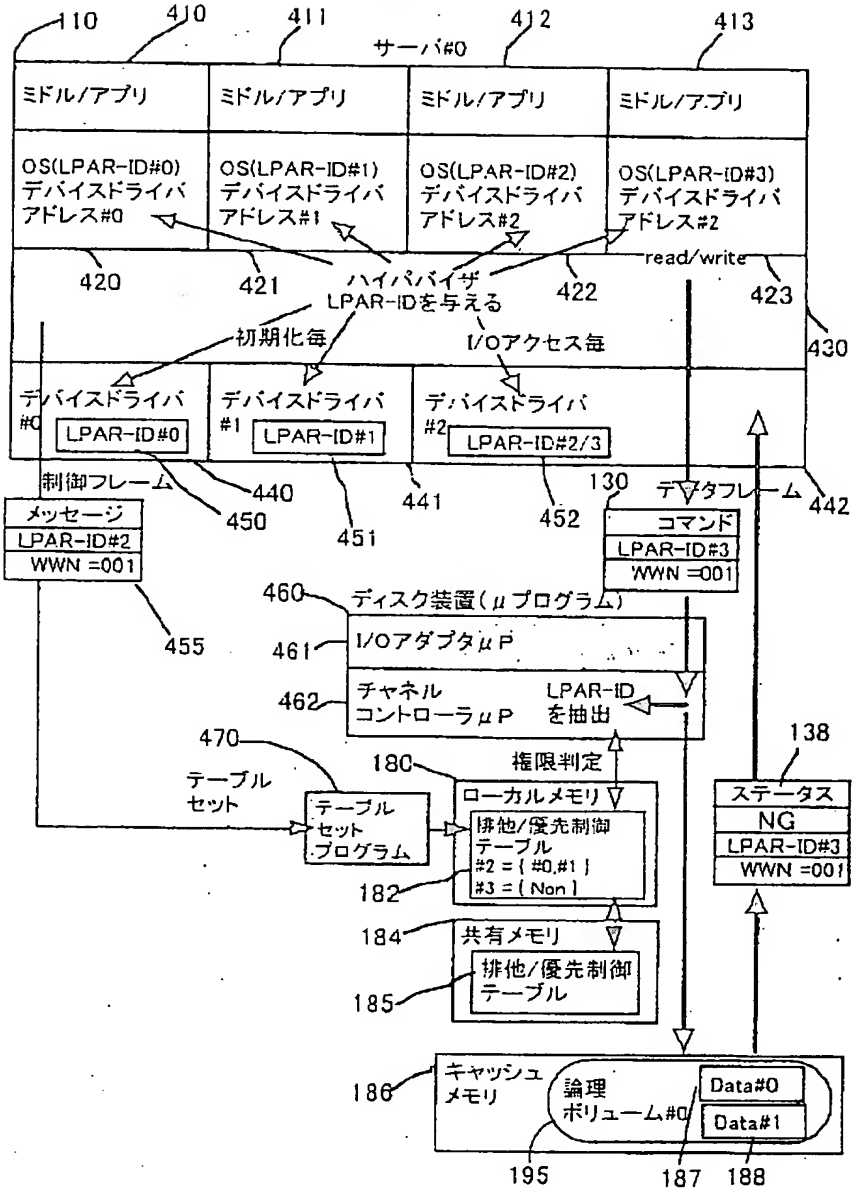
【図４】

図4



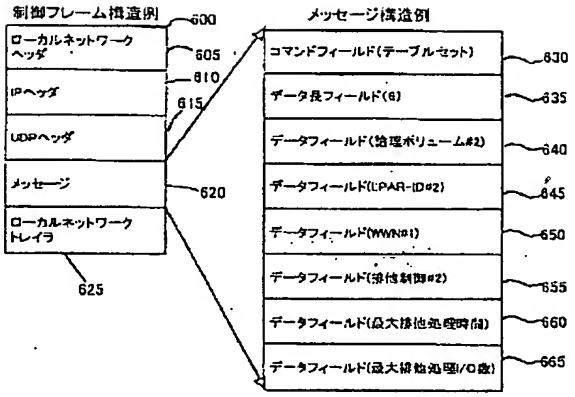
【図５】

図5



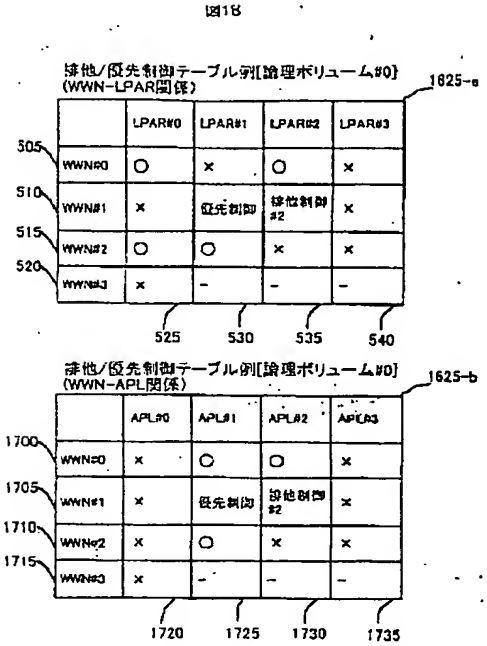
【図７】

図7



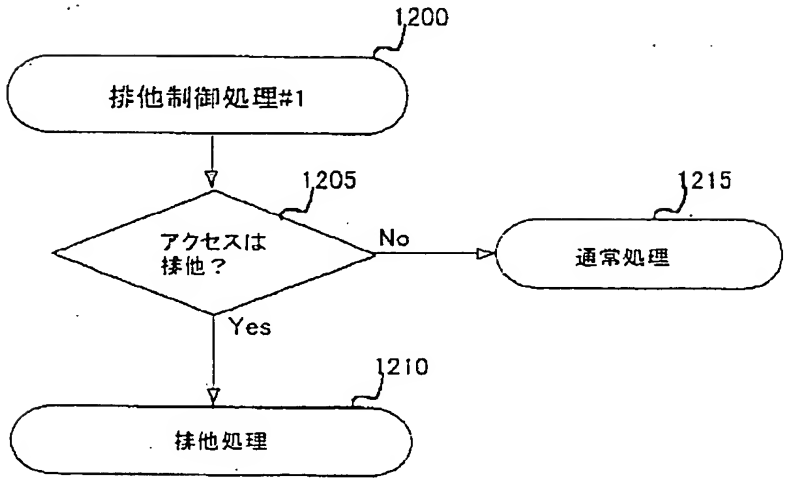
【図１８】

図18



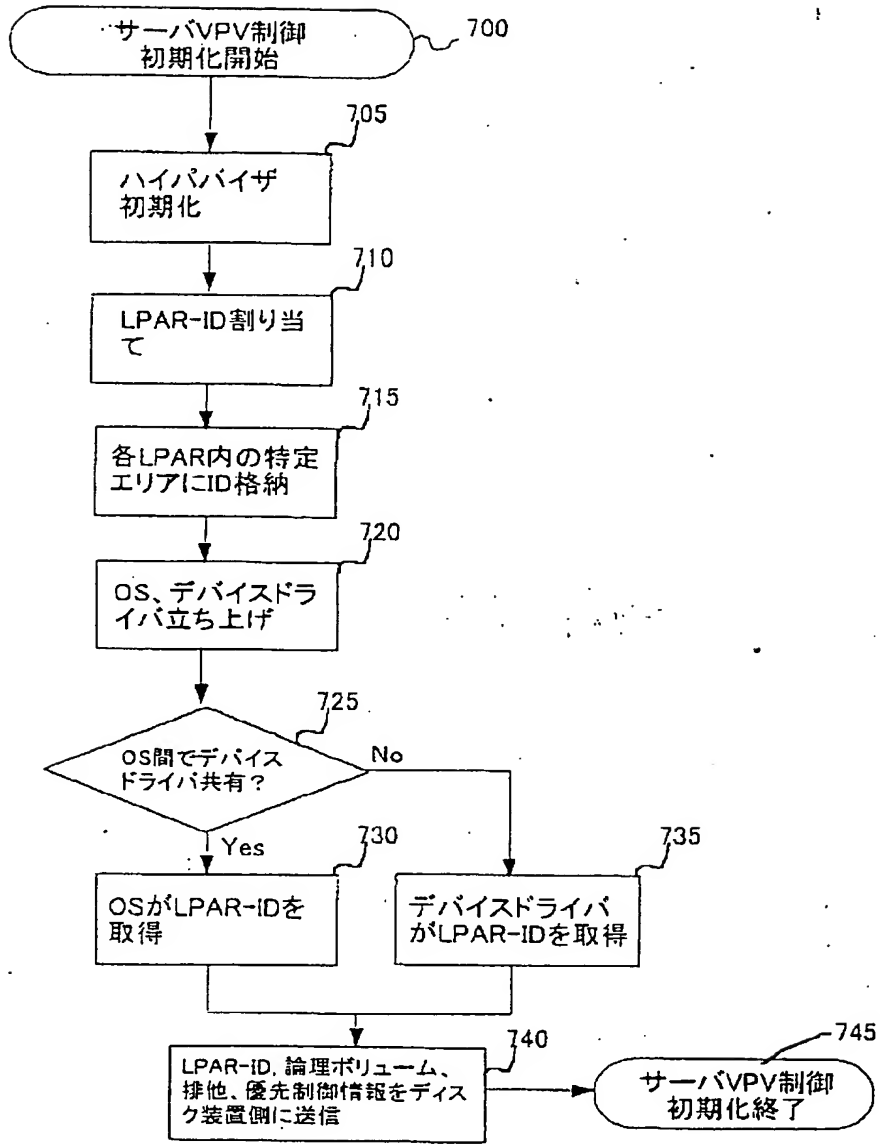
【図１３】

図13



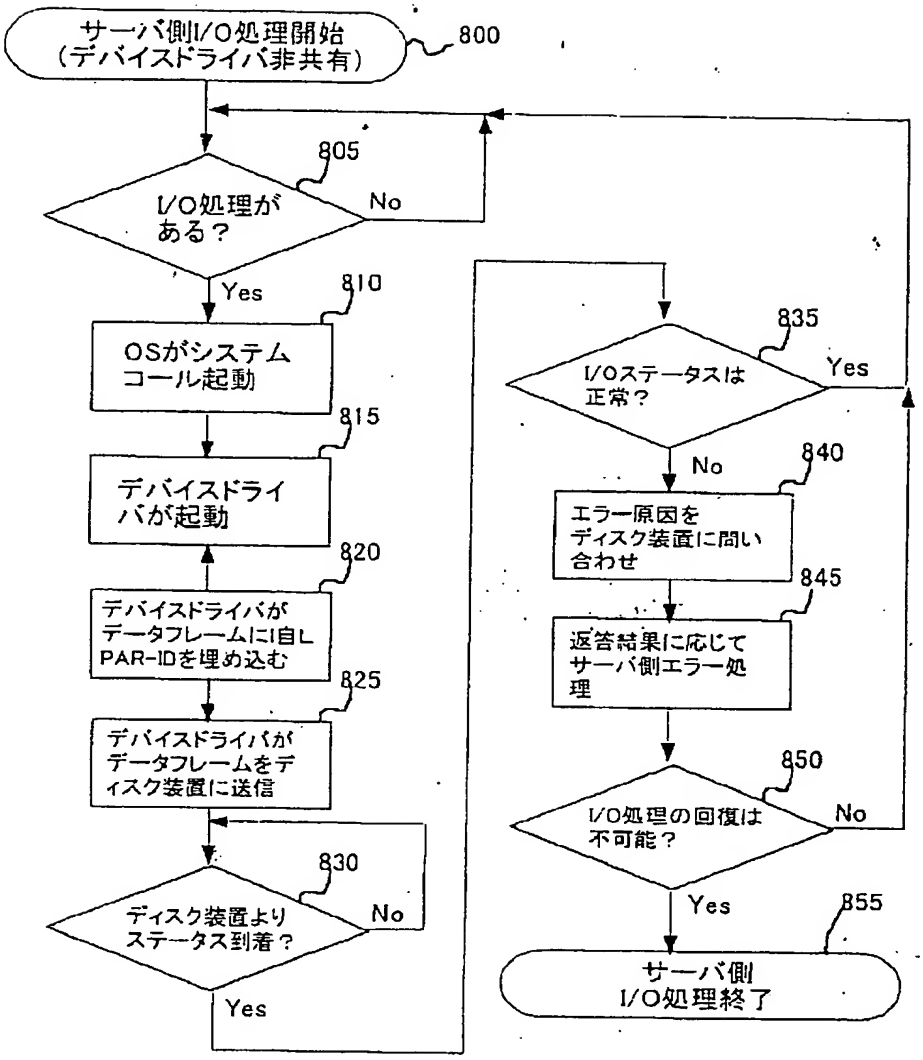
【図８】

図８



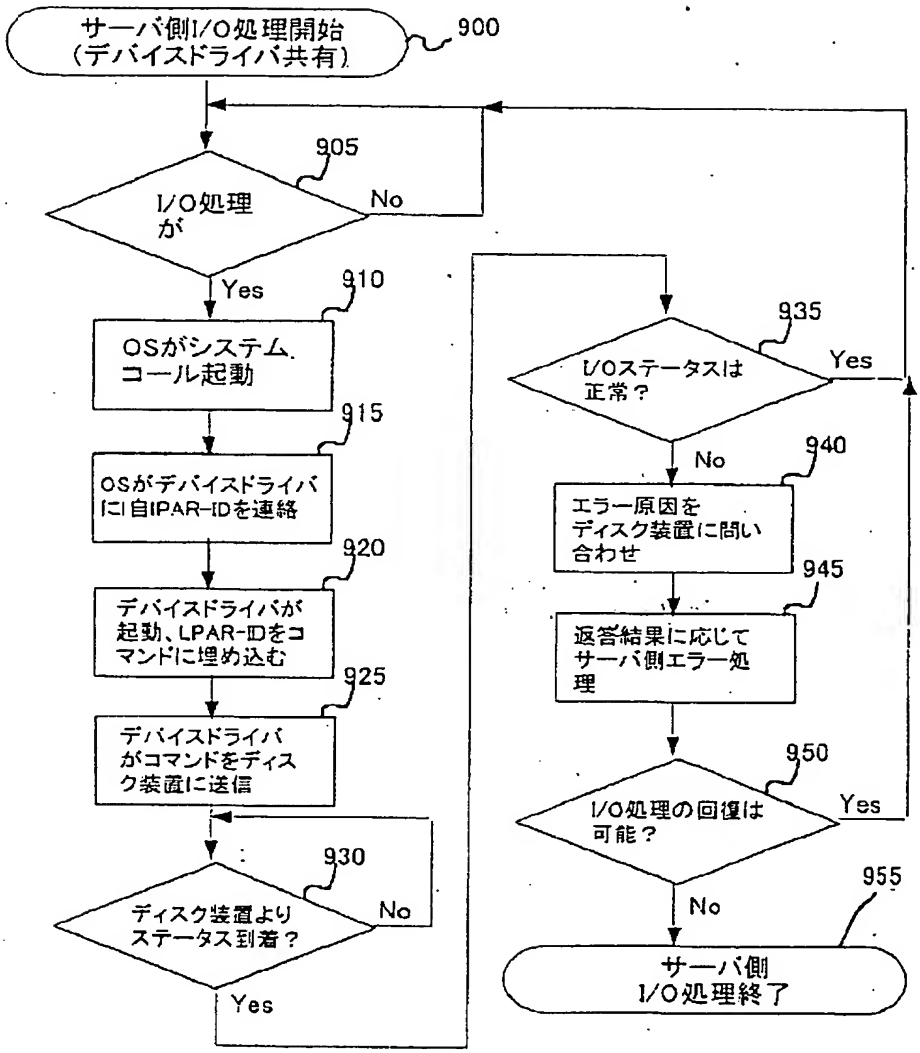
【図９】

図9



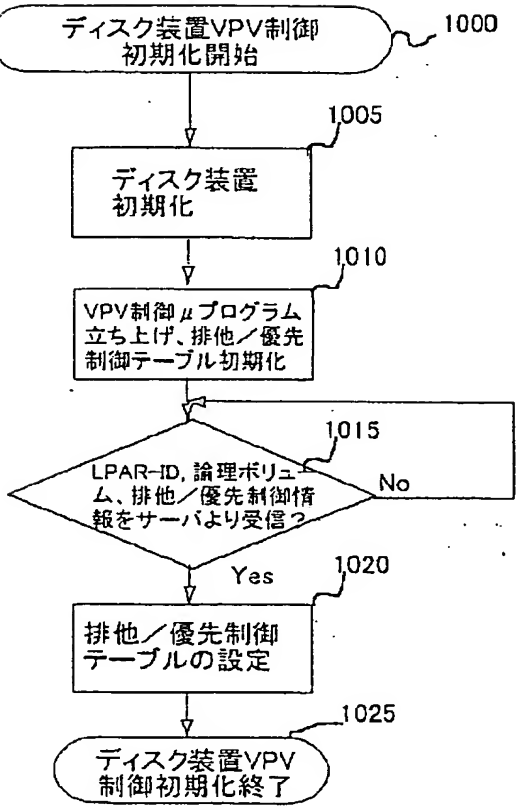
【図１０】

図10



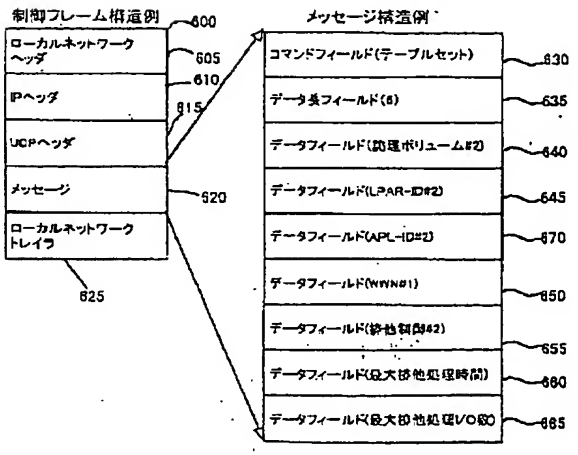
【図１１】

図11



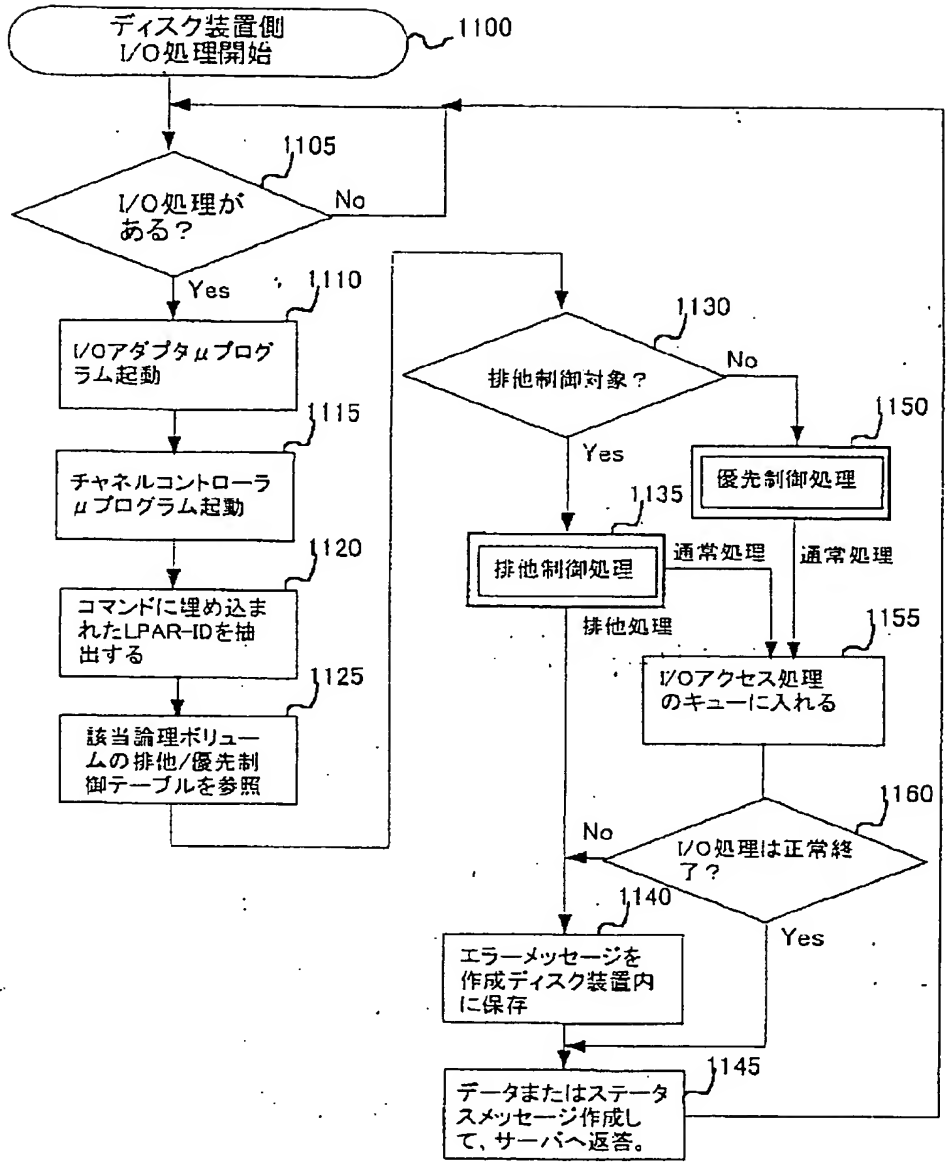
【図２０】

図20



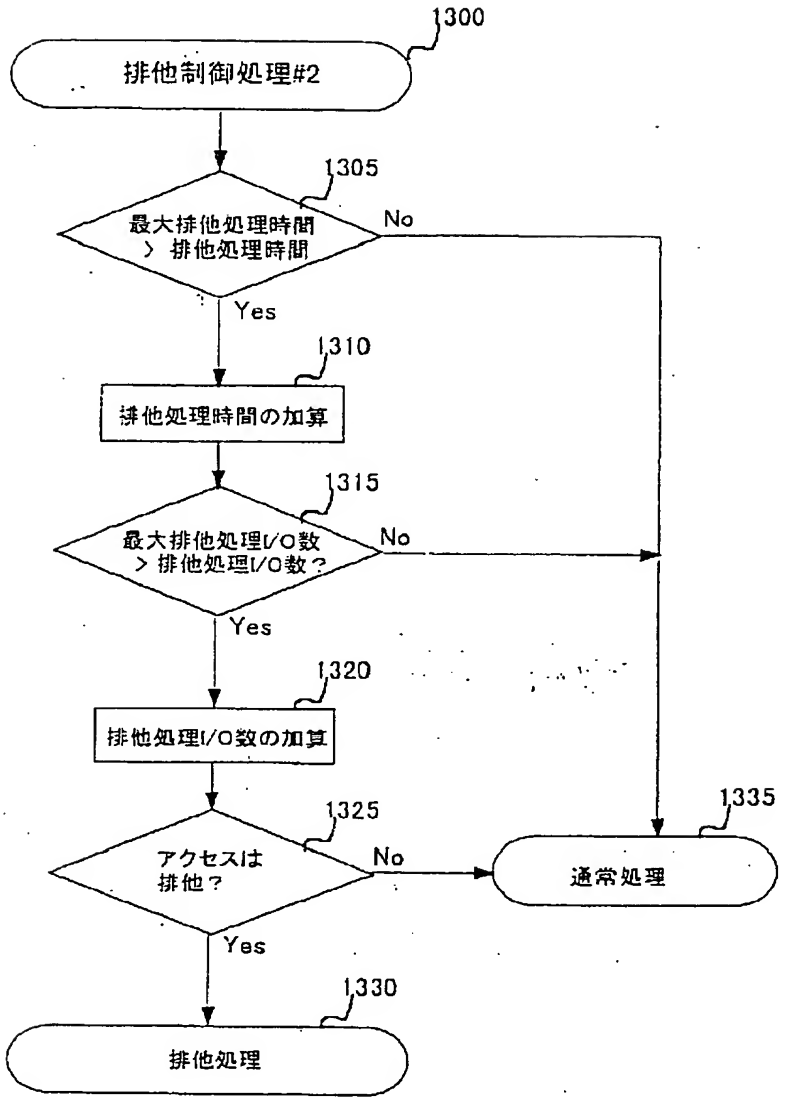
【図１２】

図12



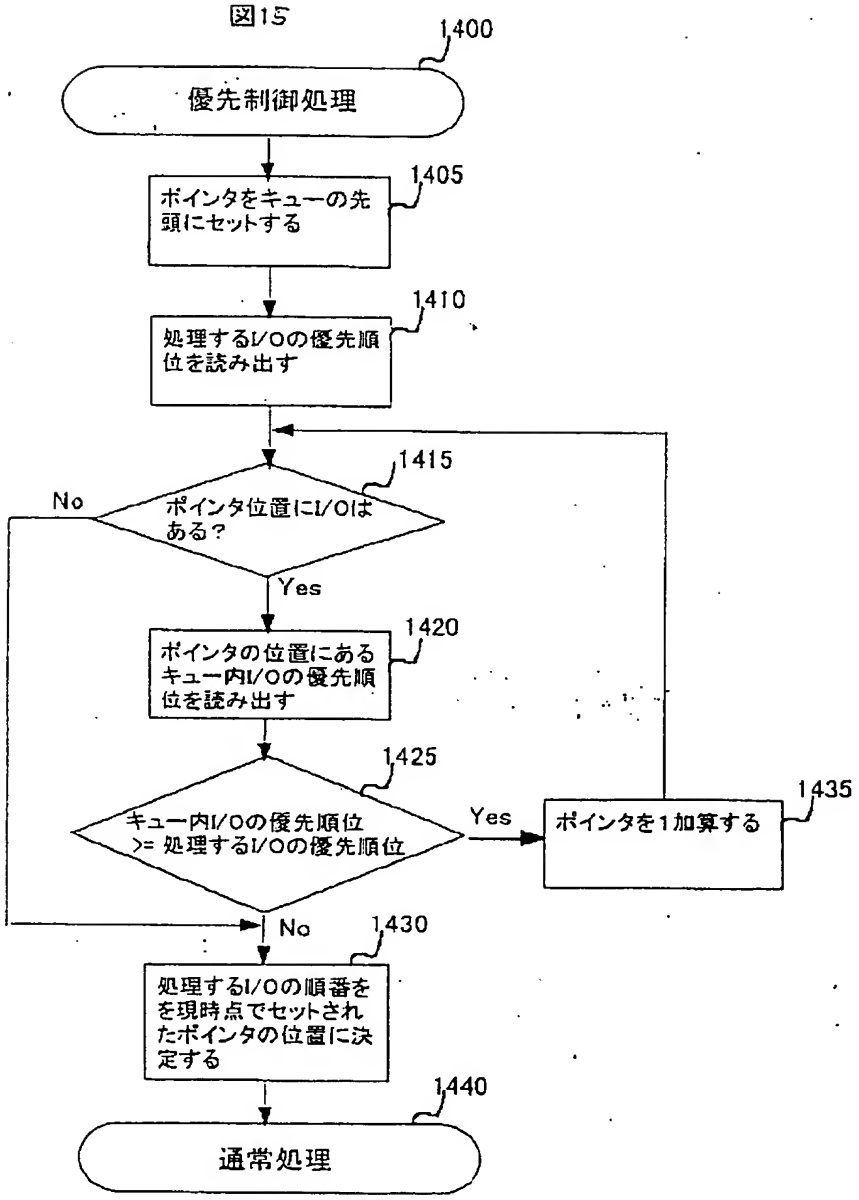
【図１４】

図14-

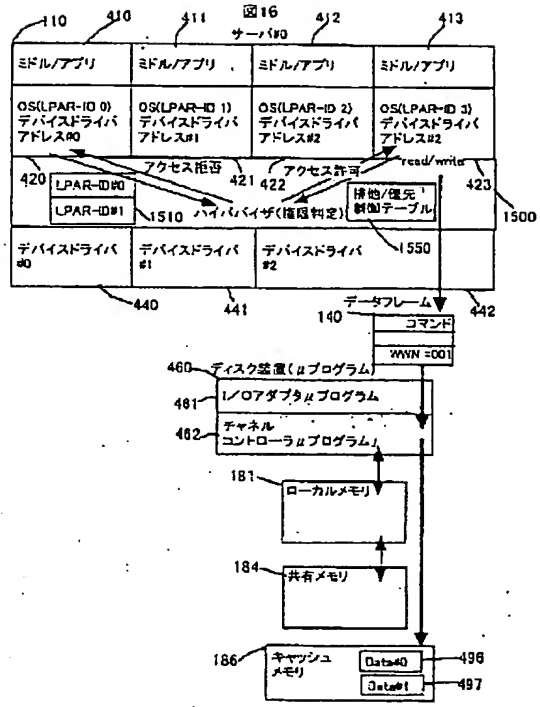


【図１５】

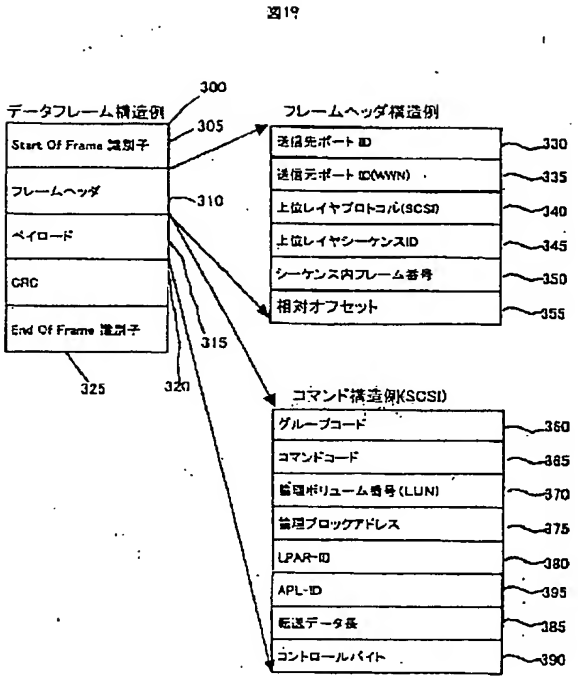
図15



【図１６】

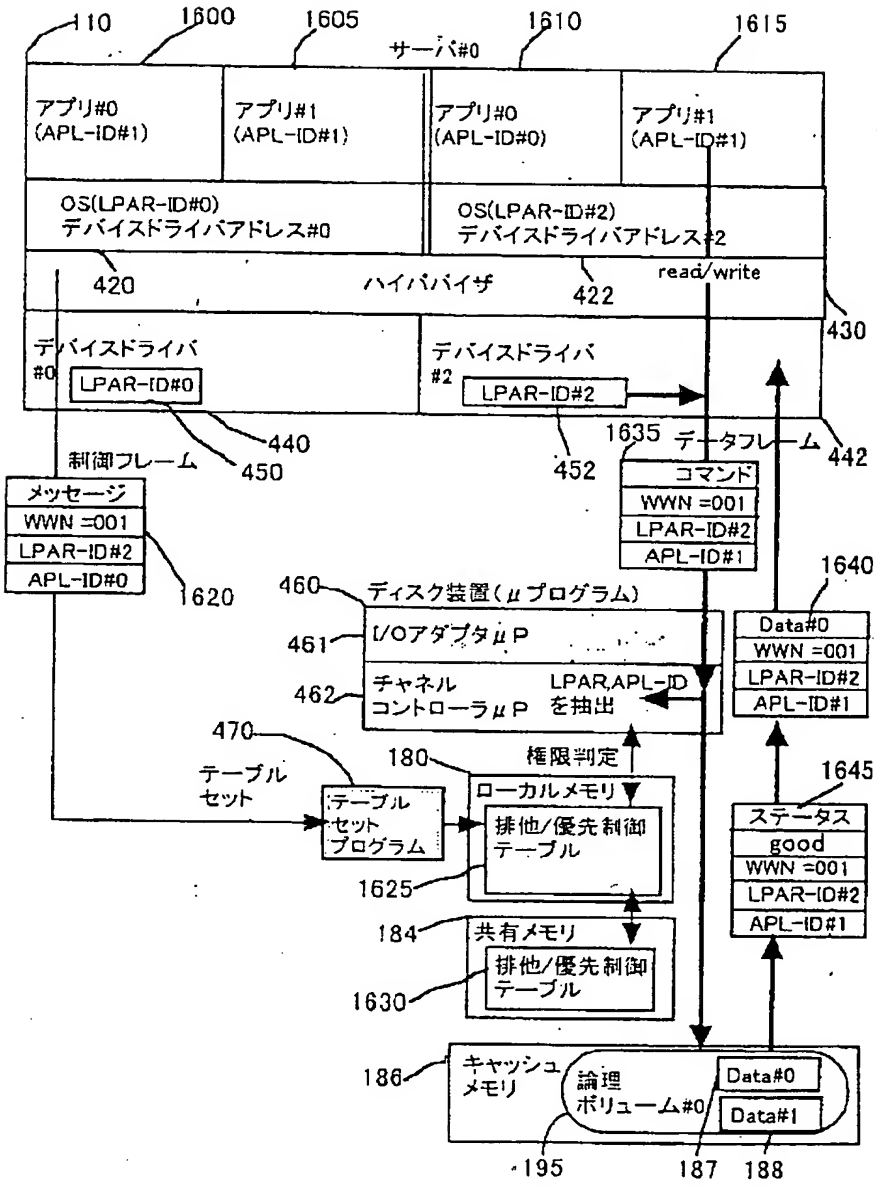


【図１９】



【図１７】

図17



フロントページの続き

(72)発明者　小田原　宏明
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

Ｆターム(参考)　5B045 EE06 EE08 EE11
5B082 EA11 JA03